

Algebraic Soft-Decision Decoding of Hermitian Codes

Kwankyu Lee and Michael E. O'Sullivan

Abstract

An algebraic soft-decision decoder for Hermitian codes is presented. We apply Koetter and Vardy's soft-decision decoding framework, now well established for Reed-Solomon codes, to Hermitian codes. First we provide an algebraic foundation for soft-decision decoding. Then we present an interpolation algorithm finding the Q -polynomial that plays a key role in the decoding. With some simulation results, we compare performances of the algebraic soft-decision decoders for Hermitian codes and Reed-Solomon codes, favorable to the former.

Index Terms

Hermitian codes, algebraic soft-decision decoding, interpolation algorithm, Gröbner bases.

I. INTRODUCTION

Sudan and Guruswami's list decoding of Reed-Solomon codes [1], [2] has developed into algebraic soft-decision decoding by Koetter and Vardy [3]. As Reed-Solomon codes are widely used in coding applications, algebraic soft-decision decoding is regarded as one of the most important developments for Reed-Solomon codes. Hence there have been many subsequent works to make the decoding method efficient and practical [4], [5], [6], [7], [8], [9]. Engineers have proposed fast electronic circuits implementing the algebraic soft-decision decoder [10], [11], [12]. One may say that the algebraic soft-decision decoding of Reed-Solomon codes is now in a mature state for deployment in applications [13].

Reed-Solomon codes are the simplest algebraic geometry codes [14]. Therefore it is natural that the list decoding of Reed-Solomon codes was soon extended to algebraic geometry codes by Shokrollahi and Wasserman [15] and Guruswami and Sudan [2]. However, it seems that no algebraic geometry codes other than Reed-Solomon codes have been considered for algebraic soft-decision decoding. One reason for this unbalanced situation is presumably that the complexity of an algebraic soft-decision decoder for algebraic geometry codes would be prohibitively huge as the complexity for Reed-Solomon codes is already very large. However, algebraic geometry codes have the advantage that they are longer than Reed-Solomon codes over the alphabet of the same size, promising better performance. We may also expect that once we have an explicit formulation of algebraic soft-decision decoding for algebraic geometry codes, some clever ways to reduce the complexity to a practical level may be found, as has happened for Reed-Solomon codes [4].

In this work, we present an algebraic soft-decision decoder for Hermitian codes. Hermitian codes are one of the best studied algebraic geometry codes, and they are often regarded as the first candidate among algebraic geometry codes that could compete with Reed-Solomon codes. To formulate an algebraic soft-decision decoder for Hermitian codes, we basically follow the path set out by Koetter and Vardy for Reed-Solomon codes. Thus there are three main steps of the decoding: the multiplicity assignment step, the interpolation step, and the root-finding step. For the multiplicity assignment step and the root-finding step, we may use algorithms in [3] and [16], respectively. Here we focus on the interpolation step, the goal of which is to construct the Q -polynomial whose roots give the candidate codewords. As for mathematical contents, this work is an extension of our previous [17] and [18]. The core contribution of the present work is an algorithm constructing a set of generators of a certain module from which we extract the Q -polynomial using the Gröbner conversion algorithm given in [17].

K. Lee is with the Department of Mathematics, Chosun University, Gwangju 501-759, Korea (e-mail: kwankyu@chosun.ac.kr).

M. E. O'Sullivan is with the Department of Mathematics and Statistics, San Diego State University, San Diego, CA 92182-7720, USA (e-mail: mosulliv@math.sdsu.edu).

This work was supported by research funds from Chosun University, 2008.

In Section 2, we review the definitions of basic concepts and the properties of Hermitian curves and codes. We refer to [19] and [14] for the basic theory of algebraic curves and algebraic geometry codes, and [20] and [21] for Gröbner bases and commutative algebra. In Section 3, we formulate the algebraic soft-decision decoding of Hermitian codes. We present our interpolation algorithm in Section 4 and a complexity analysis of the decoding algorithm in Section 5. In Section 6, we provide some simulation results of the algebraic soft-decision decoder. As this work is an extension of [17], we omitted some proofs that can be found in that work but allowed some similar materials included here for exposition purposes.

II. PRELIMINARIES

A. Hermitian curves

Let q be a prime power, and let \mathbb{F} denote a finite field with q^2 elements. The Hermitian curve $H \subset \mathbb{A}_{\mathbb{F}}^2$ is the affine plane curve defined by the absolutely irreducible polynomial $Y^q + Y - X^{q+1}$ over \mathbb{F} . The coordinate ring of H is the integral domain

$$R = \mathbb{F}[X, Y] / \langle Y^q + Y - X^{q+1} \rangle = \mathbb{F}[x, y],$$

with x and y denoting the residue classes of X and Y , respectively. Note that every element of R can be written uniquely as a polynomial of x and y with y -degree less than q , as we have $y^q + y - x^{q+1} = 0$. So R is also a free module of rank q over $\mathbb{F}[x]$. The function field $K(H)$ is the quotient field of R .

For each $\alpha \in \mathbb{F}$, there are exactly q elements $\beta \in \mathbb{F}$ such that $\text{Tr}_{\mathbb{F}/\mathbb{F}_q}(\beta) = \beta^q + \beta = \alpha^{q+1}$. Therefore there are q^3 rational points P_1, P_2, \dots, P_n of H with $n = q^3$, which can be grouped into q^2 classes of q points with the same x -coordinates. A rational point P of H is associated with a maximal ideal $\mathfrak{m}_P = \{f \in R \mid f(P) = 0\}$, and the local ring \mathcal{O}_P of H at P is the localization of R at \mathfrak{m}_P . For a nonzero $f \in R$, the valuation $v_P(f)$ is the largest integer r such that $f \in \mathfrak{m}_P^r$.

The projective closure of H is a smooth curve with a unique rational point P_∞ at infinity. The functions x and y on H have poles at P_∞ of orders q and $q+1$, respectively, that is, $v_{P_\infty}(x) = -q$ and $v_{P_\infty}(y) = -q-1$. The genus of H is given by $g = q(q-1)/2$. It is well known that the number of rational points of the curve H attains the maximum value possible for the genus and the size of the base field.

B. Hermitian codes

For $u \geq 0$, the \mathbb{F} -linear space $\mathcal{L}(uP_\infty) = \{f \in K(H) \mid (f) + uP_\infty \geq 0\}$ has a basis consisting of $x^i y^j$ for $0 \leq i$, $0 \leq j \leq q-1$, and $qi + (q+1)j \leq u$. Therefore

$$R = \bigcup_{u=0}^{\infty} \mathcal{L}(uP_\infty) = \bigoplus_{\substack{0 \leq i \\ 0 \leq j \leq q-1}} \mathbb{F} \cdot x^i y^j.$$

Recall that the Hamming space \mathbb{F}^n is an \mathbb{F} -linear space with the Hamming distance function d . For $1 \leq i \leq n$, let $P_i = (\alpha_i, \beta_i)$. The evaluation map $\text{ev} : R \rightarrow \mathbb{F}^n$ defined by

$$\varphi \mapsto (\varphi(P_1), \varphi(P_2), \dots, \varphi(P_n))$$

is a linear map over \mathbb{F} . We now fix a positive integer u . The Hermitian code C_u is defined to be the image of $\mathcal{L}(uP_\infty)$ by the evaluation map. If $u < n$, then ev is injective on $\mathcal{L}(uP_\infty)$, and the dimension of C_u is equal to $\dim_{\mathbb{F}}(\mathcal{L}(uP_\infty))$, which is $u+1-g$ for $u \geq 2g-1$ by the Riemann-Roch theorem. Note also that the minimum distance of C_u is at least $n-u$.

Define

$$h_i = -\frac{(x^{q^2} - x)(y^q + y - \beta_i^q - \beta_i)}{(x - \alpha_i)(y - \beta_i)} \in R$$

for $1 \leq i \leq n$. For a vector $v = (v_1, v_2, \dots, v_n) \in \mathbb{F}^n$, define

$$h_v = \sum_{i=1}^n v_i h_i.$$

We can easily prove that $h_i(P_j) = 1$ if $j = i$, and 0 otherwise. Therefore $\text{ev}(h_v) = v$ for all $v \in \mathbb{F}^n$.

Example 1. Let $q = 2$ and $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$. We consider the Hermitian curve H defined by $Y^2 + Y + X^3$ over \mathbb{F}_4 . There are 8 rational points on H ,

$$(0, 0), (0, 1), (1, \alpha), (1, \alpha^2), (\alpha, \alpha), (\alpha, \alpha^2), (\alpha^2, \alpha), (\alpha^2, \alpha^2).$$

Let $u = 4$. The linear space $\mathcal{L}(4P_\infty)$ is spanned by the basis $\{1, x, y, x^2\}$. Hermitian code C_4 is a linear code over \mathbb{F}_4 of length 8 and dimension 4. We use the following generator matrix for encoding

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & \alpha^2 & \alpha \\ 0 & 1 & 0 & 1 & 0 & 1 & \alpha & \alpha^2 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Note that the positions 1, 2, 3, 5 form an information set of G . Our message is $(1, \alpha^2, 0, \alpha)$, which is encoded into the codeword

$$(1, \alpha^2, 0, \alpha)G = (1, \alpha^2, 0, \alpha, \alpha, 0, 0, \alpha).$$

The functions h_i are as follows:

$$\begin{aligned} h_1 &= (x^3 + 1)y + x^3 + 1, \\ h_2 &= (x^3 + 1)y, \\ h_3 &= (x^3 + x^2 + x)y + \alpha^2 x^3 + \alpha^2 x^2 + \alpha^2 x, \\ h_4 &= (x^3 + x^2 + x)y + \alpha x^3 + \alpha x^2 + \alpha x, \\ h_5 &= (x^3 + \alpha x^2 + \alpha^2 x)y + \alpha^2 x^3 + x^2 + \alpha x, \\ h_6 &= (x^3 + \alpha x^2 + \alpha^2 x)y + \alpha x^3 + \alpha^2 x^2 + x, \\ h_7 &= (x^3 + \alpha^2 x^2 + \alpha x)y + \alpha^2 x^3 + \alpha x^2 + x, \\ h_8 &= (x^3 + \alpha^2 x^2 + \alpha x)y + \alpha x^3 + x^2 + \alpha^2 x. \end{aligned}$$

We will continue this example throughout.

C. Local multiplicity of curves on a surface

The smooth surface $V = H \times \mathbb{A}_{\mathbb{F}}^1$ has the coordinate ring $A(V) = R \otimes \mathbb{F}[z] = R[z]$. The function field $K(V)$ is the quotient field of $A(V)$. A rational point S of V is a pair (P_i, γ) with $1 \leq i \leq n$ and $\gamma \in \mathbb{F}$, and is associated with a maximal ideal $\mathfrak{m}_S = \{f \in A(V) \mid f(S) = 0\}$. The local ring \mathcal{O}_S of V at S is the localization of $A(V)$ at \mathfrak{m}_S . A nonzero function $f \in A(V)$ defines a curve on the surface V . The multiplicity of f at a rational point S , denoted $\text{mult}_S(f)$, is the largest integer r such that $f \in \mathfrak{m}_S^r$. We note the following properties of multiplicity on the surface V . Let P be a rational point of H .

- (i) If $f \in R$, then $v_P(f) = \text{mult}_{(P, \gamma)}(f)$ for every $\gamma \in \mathbb{F}$.
- (ii) For $f \in R$, $\text{mult}_{(P, \gamma)}(z - f) = 1$ if $f(P) = \gamma$, and 0 otherwise.
- (iii) For $f, g \in A(V)$, $\text{mult}_S(fg) = \text{mult}_S(f) + \text{mult}_S(g)$ for every rational point S of V .

III. ALGEBRAIC SOFT-DECISION DECODING

Suppose that some codeword of C_u was sent through a noisy channel. The output of the channel is some probabilistic information, for each location $1 \leq i \leq n$, of the plausibility of each $\gamma \in \mathbb{F}$. The multiplicity assignment step translates the information to a doubly indexed list

$$M = [m_{i\gamma} \mid 1 \leq i \leq n, \gamma \in \mathbb{F}]$$

of nonnegative integers, where we regard $m_{i\gamma}$ as assigned to the point $(P_i, \gamma) \in H \times \mathbb{A}_{\mathbb{F}}^1$. The integer value $m_{i\gamma}$ would be chosen roughly proportional to the plausibility of the symbol γ according to the channel output. We call M the multiplicity matrix.

Corresponding to M , define

$$I_M = \{f \in R[z] \mid \text{mult}_{(P_i, \gamma)}(f) \geq m_{i\gamma} \text{ for } 1 \leq i \leq n, \gamma \in \mathbb{F}\},$$

an ideal of $R[z]$. We call I_M the interpolation ideal. Note that by definition

$$I_M = \bigcap_{1 \leq i \leq n, \gamma \in \mathbb{F}} \mathfrak{m}_{(P_i, \gamma)}^{m_{i\gamma}}.$$

For a vector $v = (v_1, v_2, \dots, v_n) \in \mathbb{F}^n$, the score of v with respect to M is defined as

$$\text{score}_M(v) = \sum_{i=1}^n m_{iv_i}$$

Hence $\text{score}_M(v)$ is also the sum of the multiplicities of the points through which the curve $z - h_v$ passes. The task of the algebraic soft-decision decoder is to find the codeword that has the best score with respect to M . This codeword is presumed to be the most likely to have been sent, given the channel output.

Example 2 (continued). Suppose that the codeword in the previous example is sent through a noisy channel and received data gives rise to the matrix of the plausibilities of symbols

$$\begin{bmatrix} 0.604 & 0.001 & 0.171 & 0.001 & 0.567 & 0.949 & 0.997 & 0.486 \\ 0.396 & 0.158 & 0.760 & 0.000 & 0.103 & 0.010 & 0.003 & 0.022 \\ 0.000 & 0.005 & 0.013 & 0.985 & 0.279 & 0.041 & 0.000 & 0.470 \\ 0.000 & 0.836 & 0.056 & 0.014 & 0.051 & 0.000 & 0.000 & 0.022 \end{bmatrix}$$

which is then translated to the multiplicity matrix

$$M = \begin{bmatrix} 3 & 0 & 0 & 0 & 2 & 4 & 5 & 2 \\ 2 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 1 & 0 & 0 & 2 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where the rows are indexed by $\gamma = 0, 1, \alpha, \alpha^2$ from top to bottom. Note that neither of the vectors $(0, \alpha^2, 1, \alpha, 0, 0, 0, 0)$ and $(0, \alpha^2, 1, \alpha, 0, 0, 0, \alpha)$ that have the best score with respect to M is a codeword of C_4 .

Lemma 1. Let $M = [m_{i\gamma}]$ be the multiplicity matrix. Then

$$\dim_{\mathbb{F}} R[z]/I_M = \sum_{i=1}^n \sum_{\gamma \in \mathbb{F}} \binom{m_{i\gamma} + 1}{2}.$$

Proof: Since I_M is a zero-dimensional ideal,

$$\begin{aligned} \dim_{\mathbb{F}} R[z]/I_M &= \sum_{i=1}^n \sum_{\gamma \in \mathbb{F}} \dim_{\mathbb{F}} \mathcal{O}_{(P_i, \gamma)}/I_M \mathcal{O}_{(P_i, \gamma)} \\ &= \sum_{i=1}^n \sum_{\gamma \in \mathbb{F}} \dim_{\mathbb{F}} \hat{\mathcal{O}}_{(P_i, \gamma)}/I_M \hat{\mathcal{O}}_{(P_i, \gamma)}, \end{aligned}$$

where $\hat{\mathcal{O}}$ denotes the completion of the local ring. If t is a uniformizing parameter of P_i and $s = z - \gamma$, then $\hat{\mathcal{O}}_{(P_i, \gamma)}$ is isomorphic to $\mathbb{F}[[s, t]]$. So $\hat{\mathcal{O}}_{(P_i, \gamma)}/I_M \hat{\mathcal{O}}_{(P_i, \gamma)}$ is isomorphic to $\mathbb{F}[[s, t]]/(s, t)^{m_{i\gamma}}$. The conclusion follows. \square

Lemma 2. Let $\mu \in R$ with $v = \text{ev}(\mu)$. Then

$$\dim_{\mathbb{F}} R[z]/(I_M + \langle z - \mu \rangle) = \text{score}(v).$$

Proof: As in the previous proof,

$$\dim_{\mathbb{F}} R[z]/(I_M + \langle z - \mu \rangle) = \sum_{i=1}^n \sum_{\gamma \in \mathbb{F}} \dim_{\mathbb{F}} \hat{\mathcal{O}}_{(P_i, \gamma)}/(I_M + \langle z - \mu \rangle) \hat{\mathcal{O}}_{(P_i, \gamma)}.$$

Let t be a uniformizing parameter of P_i and $s = z - \gamma$ again. We find that if $\mu(P_i) = \gamma$, then $\hat{\mathcal{O}}_{(P_i, \gamma)}/(I_M + \langle z - \mu \rangle) \hat{\mathcal{O}}_{(P_i, \gamma)}$ is isomorphic to $\mathbb{F}[[s, t]]/(\langle s, t \rangle^{m_{i\gamma}} + \langle s - tu \rangle) = \mathbb{F}[[t]]/\langle t^{m_{i\gamma}} \rangle$, but collapses to the zero ring otherwise. Here u is some unit in $\hat{\mathcal{O}}_{(P_i, \gamma)}$. The conclusion follows. \square

For $f = \psi_a z^a + \dots + \psi_1 z + \psi_0 \in R[z]$ with $\psi_i \in R$, the u -weighted degree of f is defined as

$$\deg_u(f) = \max_{0 \leq i \leq a} (-v_{P_\infty}(\psi_i) + ui).$$

For $f \in R[z]$ and $\varphi \in R$, denote by $f(\varphi)$ the element in R that is obtained by substituting z with φ in f . Observe that if $\varphi \in \mathcal{L}(uP_\infty)$, then $-v_{P_\infty}(f(\varphi)) \leq \deg_u(f)$. The algebraic soft-decision decoding of Hermitian codes rests upon the following

Proposition 3. Suppose $f \in I_M$ is nonzero. If a codeword $c = \text{ev}(\mu)$ of C_u with $\mu \in \mathcal{L}(uP_\infty)$ satisfies

$$\text{score}_M(c) > \deg_u(f),$$

then $f(\mu) = 0$.

Proof: Assume that $f(\mu)$ is not zero in R . Then

$$\begin{aligned} \deg_u(f) &\geq -v_{P_\infty}(f(\mu)) \\ &= \dim_{\mathbb{F}}(R/f(\mu)) \\ &= \dim_{\mathbb{F}}(R[z]/\langle f, z - \mu \rangle) \\ &\geq \dim_{\mathbb{F}}(R[z]/(I_M + \langle z - \mu \rangle)) = \text{score}(c). \end{aligned}$$

For the first equality, see Lemma 5 in [17]. This implies that if $\text{score}_M(c) > \deg_u(f)$, we must have $f(\mu) = 0$. \square

In the interpolation step, the decoder picks a polynomial $f \in I_M$. Then by Proposition 3, all codewords whose score with respect to M is big enough can be obtained from the roots of f over R . Thus the decoder can find among the candidates the codeword that has the best score with respect to M . It should be noted that for the best performance of algebraic soft-decision decoding, it is crucial for the decoder to find a polynomial in I_M with the smallest u -weighted degree. Having the same weighted degree, the one with smaller degree in z is preferred because this reduces the work of the root-finding step. Here the idea of Gröbner bases is relevant.

We call the elements in the set

$$\Omega = \{x^i y^j z^k \mid 0 \leq i, 0 \leq j \leq q-1, 0 \leq k\}$$

monomials of $R[z]$. Recall that every element of $R[z]$ can be written as a unique linear combination over \mathbb{F} of monomials of $R[z]$. Note that

$$\deg_u(x^i y^j z^k) = qi + (q+1)j + uk.$$

For two monomials $x^{i_1} y^{j_1} z^{k_1}, x^{i_2} y^{j_2} z^{k_2}$ in Ω , we declare

$$x^{i_1} y^{j_1} z^{k_1} >_u x^{i_2} y^{j_2} z^{k_2}$$

if $\deg_u(x^{i_1} y^{j_1} z^{k_1}) > \deg_u(x^{i_2} y^{j_2} z^{k_2})$ or $k_1 > k_2$ when tied. It is easy to verify that $>_u$ is a total order on Ω . Notions such as the leading term and the leading coefficient of $f \in R[z]$ are defined in the usual way. For $f \in R[z]$, the z -degree of f , written $z\text{-deg}(f)$, is the degree of f as a polynomial in z over R .

Now we define the Q -polynomial of I_M as the unique, up to a constant multiple, element in I_M with the smallest leading term with respect to $>_u$. By the definition, the Q -polynomial is an element of I_M with the smallest u -weighted degree, and moreover it has the smallest z -degree among such elements. Therefore we may say that the Q -polynomial is an optimal choice for the interpolation step.

The last step of algebraic soft-decision decoding is to compute roots of the Q -polynomial over R or the function field $K(H)$. Only those roots that belong to $\mathcal{L}(uP_\infty)$ yield candidate codewords. If the list of the candidate codewords is empty, the decoder may declare decoding failure or resort to hard-decision decoding directly from the channel output. If there are several codewords in the list, then the decoder chooses the codeword that has the best score, and outputs the received message by projecting the codeword on the information set.

Example 3 (continued). The Q -polynomial of I_M

$$\begin{aligned} Q = & z^5 + (\alpha^2 x^3 + \alpha xy + x^2 + \alpha y)z^4 + (\alpha x^4 y + \alpha^2 x^5 + x^3 + xy + x^2 + y + \alpha^2 x + 1)z^3 \\ & + (\alpha^2 x^6 y + \alpha x^7 + \alpha^2 x^5 y + \alpha^2 x^6 + \alpha^2 x^4 y + \alpha^2 x^5 + \alpha^2 x^3 y + x^4 + x^3 + \alpha xy + \alpha^2 x)z^2 \\ & + (x^8 y + \alpha^2 x^9 + \alpha x^8 + \alpha^2 x^7 + x^6 + \alpha x^4 y + x^5 + \alpha x^3 y + \alpha x^4 + \alpha^2 x^3 + \alpha^2 xy + \alpha x^2 + \alpha^2 y)z \\ & + \alpha^2 x^{11} + x^{10} + x^8 y + \alpha x^9 + \dots + \alpha^2 x^2 y + x^3 + \alpha^2 xy + y \end{aligned} \quad (1)$$

is obtained by the interpolation algorithm in the next section. It turns out the Q -polynomial has the factorization

$$Q = (z + x^2 + \alpha^2 y + x) \left(z + \alpha^2 x^2 + \alpha y + x + 1 \right) \left(z^3 + (\alpha^2 x^3 + \alpha x y + \alpha^2 x^2 + \alpha^2 y + 1) z^2 \right. \\ \left. + (\alpha x^4 y + \alpha x^5 + \alpha x^2 y + \alpha x^3 + \alpha y + \alpha x) z + x^7 + \alpha^2 x^6 + x^3 y + x^4 + \alpha x^2 y + \alpha^2 x^3 + \alpha x y + \alpha^2 y \right).$$

Therefore a root-finding algorithm will output two roots. The first root $x^2 + \alpha^2 y + x$ gives the codeword

$$c_1 = (0, \alpha^2, 1, \alpha, 0, \alpha^2, 0, \alpha^2)$$

whose score is 22 while the second root $\alpha^2x^2 + \alpha y + x + 1$ gives the codeword

$$c_2 = (1, \alpha^2, 0, \alpha, \alpha, 0, 0, \alpha)$$

whose score is 23. Therefore the decoder chooses c_2 , and the received message is

$$(1, \alpha^2, 0, \alpha),$$

which is the correct sent message.

We will need upper bounds on the u -weighted degree and the z -degree of the Q -polynomial of I_M . Let Q denote the Q -polynomial of I_M .

Proposition 4. *If $A \subset \Omega$ is a finite set of monomials of $R[z]$ such that*

$$|A| > \sum_{i=1}^n \sum_{\gamma \in \mathbb{F}} \binom{m_{i\gamma} + 1}{2},$$

then there is a set of coefficients $c_\varphi \in \mathbb{F}$ such that $0 \neq \sum_{\varphi \in A} c_\varphi \varphi \in I_M$.

Proof: Lemma 1 implies that monomials in A are linearly dependent over \mathbb{F} in $R[z]/I_M$. On the other hand, they are linearly independent over \mathbb{F} in $R[z]$. \square

In a table, we arrange monomials of $R[z]$ such that the monomials in the same column have the same u -weighted degree and the monomials in the same row have the same z -degree. Let weighted degrees increase from left to right and z -degrees from bottom to top.

Example 4 (continued). Note that $\deg_u(x^i y^j z^k) = 2i + 3j + 4k$. So we have the following table

3													z^3	\bigcirc	\dots
2									z^2	\bigcirc	xz^2	yz^2	x^2z^2	xyz	\dots
1					z	\bigcirc	xz	yz	x^2z	xyz	x^3z	x^2yz	x^4z	x^3yz	\dots
0	1	\bigcirc	x	y	x^2	xy	x^3	x^2y	x^4	x^3y	x^5	x^4y	x^6	x^5y	\dots
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	

The symbol \bigcirc indicates that there is no monomial for the position.

The table of monomials of $R[z]$ suggests the following formula. Let $G(i) = 0$ if i is a Weierstrass gap at P_∞ , and 1 otherwise. Note that $G(i) = 1$ for $i \geq 2g$. The number of monomials with u -weighted degree i is

$$C(i) = \sum_{j=0}^{\lfloor i/u \rfloor} G(i - uj).$$

Let w be the smallest integer such that

$$N = 1 + \sum_{i=1}^n \sum_{\gamma \in \mathbb{F}} \binom{m_{i\gamma} + 1}{2} \leq \sum_{i=0}^w C(i).$$

Let $l = \lfloor w/u \rfloor$. Then the u -weighted degrees and the z -degrees of monomials up to the N th monomial are not greater than w and l , respectively. Now Proposition 4 implies $\deg_u(Q) \leq w$ and $z\text{-deg}(Q) \leq l$.

Example 5 (continued). $G(0) = 1, G(1) = 0$, and $G(i) = 1$ for $i \geq 2$ since $g = 1$. So we have

i	0	1	2	3	4	5	\dots	21	22	23	24	25	\dots
$C(i)$	1	0	1	1	2	1	\dots	5	6	6	7	6	\dots
$\sum_{j=0}^i C(j)$	1	1	2	3	5	6	\dots	66	72	78	85	91	\dots

For our M , $N = 76$. Therefore $w = 23$, $l = 5$. Hence $\deg_u(Q) \leq 23$ and $z\text{-deg}(Q) \leq 5$.

IV. AN INTERPOLATION ALGORITHM

Let l be a positive integer such that $z\text{-deg}(Q) \leq l$. Define

$$R[z]_l = \{f \in R[z] \mid z\text{-deg}(f) \leq l\}.$$

Note that $R[z]_l$ is a free module over R of rank $l + 1$ with a free basis $1, z, \dots, z^l$. Define

$$I_{M,l} = I_M \cap R[z]_l.$$

Clearly $I_{M,l}$ is a submodule of $R[z]_l$ over R .

Recall that the ring $R = \mathbb{F}[x, y]$ is in turn a free module over $\mathbb{F}[x]$ of rank q , with a free basis $\{1, y, \dots, y^{q-1}\}$. So we may view $R[z]_l$ as a free module of rank $q(l + 1)$ over $\mathbb{F}[x]$ with a free basis $\{y^j z^i \mid 0 \leq i \leq l, 0 \leq j \leq q - 1\}$. The elements of $\Omega \cap R[z]_l$ will be called monomials of $R[z]_l$. It is clear that the total order $>_u$ is precisely a monomial order on the free module $R[z]_l$ over $\mathbb{F}[x]$. We also view $I_{M,l}$ as a submodule of the free module $R[z]_l$ over $\mathbb{F}[x]$. It is immediate that the Q -polynomial of I_M is also the element of $I_{M,l}$ with the smallest leading term with respect to $>_u$. As a consequence of the definition of Gröbner bases, Q occurs as the smallest element in any Gröbner basis of the module $I_{M,l}$ over $\mathbb{F}[x]$ with respect to $>_u$.

A. Generators of the module $I_{M,l}$ over R

We begin with

Proposition 5. Let $M = [m_{i\gamma}]$ be a doubly indexed list of nonnegative integers. For each $i = 1, 2, \dots, n$, let $n_i = \max_{\gamma \in \mathbb{F}} m_{i\gamma}$, and let $N = [n_i]$. For each i with $n_i > 0$, let γ_i be such that $m_{i\gamma_i} = n_i$. Let $M' = [m'_{i\gamma}]$ where $m'_{i\gamma} = m_{i\gamma}$ for $\gamma \neq \gamma_i$ and $m'_{i\gamma_i} = m_{i\gamma_i} - 1$. Then as a module over R ,

$$I_M = (z - h)I_{M'} + J_N$$

where $J_N = \{f \in R \mid v_{P_i}(f) \geq n_i\}$ and $h \in R$ is such that $h(P_i) = \gamma_i$.

Proof: By the properties (i), (ii), (iii) of local multiplicity, it is clear that $(z - h)I_{M'} + J_N \subset I_M$. To show the reverse inclusion, let $f \in I_M$. We can write $f = (z - h)g + r$ for some $g \in R[z]$ and $r \in R$. Let $S = (P_i, \gamma_i)$. If t is a uniformizing parameter of P_i , then t and $z - \gamma_i$ form a system of parameters of \mathcal{O}_S . Recall that the completion $\hat{\mathcal{O}}_S$ is isomorphic to the power series ring $\mathbb{F}[[z - \gamma_i, t]]$. Now if $v_{P_i}(r) = c$, then in $\hat{\mathcal{O}}_S$,

$$f = (z - \gamma_i)g + t^c u$$

for some unit u in $\hat{\mathcal{O}}_S$. Since $f \in \langle z - \gamma_i, t \rangle^{n_i}$ and $z - \gamma_i, t$ are algebraically independent over \mathbb{F} , we see that $c \geq n_i$. Then as this is true for all $1 \leq i \leq n$, it follows that $r \in J_N$. Hence $f - r = (z - h)g \in I_M$. Again by the properties of local multiplicity, $g \in I_{M'}$. Thus we showed the reverse inclusion. \square

Recall the multiplicity matrix $M = [m_{i\gamma}]$. Let $l_{\max} = \max_i \{\sum_{\gamma \in \mathbb{F}} m_{i\gamma}\}$. Initially let $m_{i\gamma}^{(0)} = m_{i\gamma}$ and $n_i^{(0)} = \max_{\gamma \in \mathbb{F}} \{m_{i\gamma}\}$. Proceed inductively for $s = 0, 1, \dots, l_{\max} - 1$. Choose γ_i such that $m_{i\gamma_i}^{(s)} = n_i^{(s)}$ if $n_i^{(s)} > 0$. Let $h^{(s)} \in R$ such that $h^{(s)}(P_i) = \gamma_i$. Let

$$m_{i\gamma}^{(s+1)} = \begin{cases} m_{i\gamma}^{(s)} - 1 & \text{if } \gamma = \gamma_i, \\ m_{i\gamma}^{(s)} & \text{if } \gamma \neq \gamma_i, \end{cases}$$

$$n_i^{(s+1)} = \max_{\gamma \in \mathbb{F}} m_{i\gamma}^{(s+1)}.$$

Now let $M^{(s)} = [m_{i\gamma}^{(s)}]$ and $N^{(s)} = [n_i^{(s)}]$. Observe $m_{i\gamma}^{(l_{\max})} = 0$ for all i, γ , and therefore $I_{M^{(l_{\max})}} = R[z]$. By induction, we get

Corollary 6. For $0 \leq l$,

$$I_{M,l} = \sum_{s=0}^l J_{N^{(s)}} \prod_{0 \leq r < s} (z - h^{(r)})$$

as a module over R . Here $J_{N^{(s)}} = R$ and $h^{(s)} = 0$ for $s \geq l_{\max}$.

B. Computing generators of the module $I_{M,l}$ over $\mathbb{F}[x]$

We may view the ideal $J_N = \{f \in R \mid v_{P_i}(f) \geq n_i\}$ as a module over $\mathbb{F}[x]$. Indeed J_N is a free module of rank q over $\mathbb{F}[x]$. Thus we obtain

Algorithm B. The input is an $n \times q^2$ matrix $M = [m_{i\gamma}]$ of nonnegative integers. The output is the generators $\{g_{s,t} \mid 0 \leq s \leq l, 0 \leq t \leq q-1\}$ of $I_{M,l}$ as a module over $\mathbb{F}[x]$. Repeat steps B1 and B2 for $s = 0, 1, \dots, l$.

B1. Let $n_i = \max_{\gamma \in \mathbb{F}} m_{i\gamma}$ for $1 \leq i \leq n$. Let $L = \{1 \leq i \leq n \mid n_i \geq 1\}$. For each $i \in L$, let $\gamma_i \in \mathbb{F}$ be such that $n_i = m_{i\gamma_i}$. Set

$$g_{s,t} \leftarrow \eta_t \prod_{0 \leq r < s} (z - h^{(r)}), \quad (2)$$

for $0 \leq t \leq q-1$, where $\{\eta_0, \eta_1, \dots, \eta_{q-1}\}$ is a set of generators of $J_{N^{(s)}} = \{f \in R \mid v_{P_i}(f) \geq n_i\}$ as a module over $\mathbb{F}[x]$. When L is empty, $J_N = R$ so $\eta_t = y^t$.

B2. Set

$$h^{(s)} \leftarrow \sum_{i \in L} \gamma_i h_i$$

and for $1 \leq i \leq n, \gamma \in \mathbb{F}$, set

$$m_{i\gamma} \leftarrow \begin{cases} m_{i\gamma} - 1 & \text{if } \gamma = \gamma_i, \\ m_{i\gamma} & \text{otherwise.} \end{cases}$$

Notice that if we compute η_t by the method in the following subsection, $g_{s,t}$ has leading term $y^t z^s$ with respect to lex order $x < y < z$.

Example 6 (continued). We continue from Example 2. We show the first few steps to compute a set of generators of $I_{M,l}$ with $l = 5$ using Algorithm B. Initially $s = 0$. Then

$$n_1 = 3, n_2 = 4, n_3 = 3, n_4 = 5, n_5 = 2, n_6 = 4, n_7 = 5, n_8 = 2.$$

As we compute in Example 7,

$$J_{N^{(0)}} = \langle x^{18} + \alpha x^{17} + \alpha^2 x^{16} + x^6 + \alpha x^5 + \alpha^2 x^4, (x^{10} + x^9 + x^4 + x^3)y + \alpha^2 x^{17} + x^{16} + \dots + x^3 \rangle$$

as a module over $\mathbb{F}[x]$. So we set

$$\begin{aligned} g_{0,0} &= x^{18} + \alpha x^{17} + \alpha^2 x^{16} + x^6 + \alpha x^5 + \alpha^2 x^4, \\ g_{0,1} &= (x^{10} + x^9 + x^4 + x^3)y + \alpha^2 x^{17} + x^{16} + \dots + x^3. \end{aligned}$$

In step B2, we compute (setting $\gamma_8 = 0$ arbitrarily)

$$\begin{aligned} h^{(0)} &= 0h_1 + \alpha^2 h_2 + 1h_3 + \alpha h_4 + 0h_5 + 0h_6 + 0h_7 + 0h_8 \\ &= \alpha^2 x^2 y + \alpha^2 x y + \alpha^2 y. \end{aligned}$$

Now the matrix of $m_{i\gamma}$ is

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 1 & 3 & 4 & 1 \\ 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 1 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Going on to $s = 1$, we have

$$n_1 = 2, n_2 = 3, n_3 = 2, n_4 = 4, n_5 = 1, n_6 = 3, n_7 = 4, n_8 = 2.$$

Then

$$J_{N^{(1)}} = \langle x^{14} + \alpha x^{13} + \alpha^2 x^{12} + \dots + \alpha x^4 + \alpha^2 x^3, \alpha x^{13} + \alpha^2 x^{12} + \alpha^2 x^{11} + \dots + \alpha x^3 + \alpha^2 x^2 \rangle$$

as a module over $\mathbb{F}[x]$. Hence

$$\begin{aligned} g_{1,0} &= (x^{14} + \alpha x^{13} + \alpha^2 x^{12} + \dots + \alpha x^4 + \alpha^2 x^3)z + (\alpha^2 x^{16} + \alpha x^{15} + \alpha^2 x^4 + \alpha x^3)y \\ g_{1,1} &= (x^7 + \alpha x^6 + \alpha^2 x^5 + x^4 + \alpha x^3 + \alpha^2 x^2)yz \\ &\quad + (\alpha x^{13} + \alpha^2 x^{12} + \alpha^2 x^{11} + \dots + \alpha x^3 + \alpha^2 x^2)z \\ &\quad + (x^{15} + \alpha^2 x^{14} + x^{13} + x^{10} + x^9 + \alpha^2 x^8 + x^7 + x^4)y \\ &\quad + \alpha^2 x^{12} + \alpha x^{11} + \alpha^2 x^6 + \alpha x^5 \end{aligned}$$

Now $h^{(1)} = \alpha x^3 y + \alpha x^2 y + \alpha^2 x^3 + \alpha x^2 + \alpha^2 y + x$ and the matrix of $m_{i\gamma}$ is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 2 & 3 & 1 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 1 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Proceeding this way until $s = 5$, we obtain a set of generators of the module $I_{M,l}$. We arrange the coefficients (polynomials in x) of the generators in the following matrix

$$\begin{bmatrix} x^{18} + \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \alpha^2 x^{17} + \dots & x^{10} + \dots & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & \alpha^2 x^{16} + \dots & x^{14} + \dots & 0 & \dots & 0 & 0 & 0 & 0 \\ \alpha^2 x^{12} + \dots & x^{15} + \dots & \alpha x^{13} + \dots & x^7 + \dots & \dots & 0 & 0 & 0 & 0 \\ x^{18} + \dots & \alpha^2 x^{15} + \dots & \alpha^2 x^{13} + \dots & \alpha x^{13} + \dots & \dots & 0 & 0 & 0 & 0 \\ \alpha x^{17} + \dots & x^{14} + \dots & x^{12} + \dots & \alpha^2 x^{12} + \dots & \dots & 0 & 0 & 0 & 0 \\ \alpha x^{17} + \dots & \alpha^2 x^{17} + \dots & x^{15} + \dots & \alpha^2 x^{12} + \dots & \dots & 0 & 0 & 0 & 0 \\ \alpha x^{16} + \dots & \alpha x^{16} + \dots & \alpha^2 x^{14} + \dots & \alpha^2 x^{11} + \dots & \dots & 0 & 0 & 0 & 0 \\ \alpha x^{20} + \dots & \alpha^2 x^{17} + \dots & \alpha x^{15} + \dots & \alpha^2 x^{15} + \dots & \dots & x^3 + \dots & 0 & 0 & 0 \\ x^{18} + \dots & \alpha x^{17} + \dots & \alpha^2 x^{15} + \dots & \alpha x^{13} + \dots & \dots & \alpha^2 x & 1 & 0 & 0 \\ x^{20} + \dots & x^{20} + \dots & \alpha x^{18} + \dots & \alpha x^{15} + \dots & \dots & x^3 + \dots & x^3 + \alpha x & 1 & 0 \\ x^{23} + \dots & x^{19} + \dots & \alpha x^{18} + \dots & \alpha x^{18} + \dots & \dots & x^6 + \alpha x^4 & \alpha x^2 + x & 0 & 1 \end{bmatrix} \quad (3)$$

where the rows are $g_{0,0}, g_{0,1}, g_{1,0}, g_{1,1}, \dots, g_{5,0}, g_{5,1}$ in this order, and the columns are coefficients of $1, y, z, yz, z^2, yz^2, \dots, z^5, yz^5$ in this order.

C. Computing generators of J_N

We now tackle the task of computing a set of generators of J_N as a module over $\mathbb{F}[x]$. For this, we switch to a different indexing of the rational points of H by grouping the q^3 rational points into q^2 classes with the same x -coordinates. Thus the rational points are $P_{a,b} = (\alpha_a, \beta_{a,b})$ for $1 \leq a \leq q^2$ and $1 \leq b \leq q$. Let $\mu_{a,b} = n_i$ if $P_{a,b}$ is the point P_i . Also assume that for each $1 \leq a \leq q^2$, we have arranged the index b such that $\mu_{a,b}$ are put in decreasing order,

$$\mu_{a,1} \geq \mu_{a,2} \geq \dots \geq \mu_{a,q}.$$

With the new notations,

$$J_N = \{f \in R \mid v_{P_{a,b}}(f) \geq \mu_{a,b} \text{ for } 1 \leq a \leq q^2 \text{ and } 1 \leq b \leq q\}.$$

Proposition 7. For $1 \leq b < c \leq q$, suppose that $f_{b,c} \in \mathbb{F}[x]$ satisfy

$$v_{P_{a,b}}(y - f_{b,c}) \geq \mu_{a,b} - \mu_{a,c}$$

for all $1 \leq a \leq q^2$. Define for $c = 1, 2, \dots, q$,

$$g_c = \prod_{1 \leq a \leq q^2} (x - \alpha_a)^{\mu_{a,c}} \prod_{1 \leq b < c} (y - f_{b,c}). \quad (4)$$

Then $J_N = \langle g_1, g_2, \dots, g_q \rangle$ as a module over $\mathbb{F}[x]$.

Proof: Let $1 \leq c \leq q$. Then for $1 \leq a \leq q^2$ and $1 \leq b \leq q$,

$$\begin{aligned} v_{P_{a,b}}(g_c) &= \mu_{a,c} v_{P_{a,b}}(x - \alpha_a) + \sum_{1 \leq b' < c} v_{P_{a,b}}(y - f_{b',c}) \\ &\geq \mu_{a,c} + v_{P_{a,b}}(y - f_{b,c}) \geq \mu_{a,b}. \end{aligned}$$

Therefore $g_c \in J_N$. Recall that we may view R as a free module of rank q over $\mathbb{F}[x]$. Let J be the submodule of R generated by g_1, g_2, \dots, g_q over $\mathbb{F}[x]$. Then R/J is isomorphic to

$$\bigoplus_{1 \leq c \leq q} \mathbb{F}[x] / \langle \prod_{1 \leq a \leq q^2} (x - \alpha_a)^{\mu_{a,c}} \rangle.$$

Therefore

$$\dim_{\mathbb{F}} R/J = \sum_{1 \leq c \leq q} \dim_{\mathbb{F}} \mathbb{F}[x] / \langle \prod_{1 \leq a \leq q^2} (x - \alpha_a)^{\mu_{a,c}} \rangle = \sum_{1 \leq c \leq q} \sum_{1 \leq a \leq q^2} \mu_{a,c}.$$

On the other hand, as $J_N = \bigcap_{P_{a,b}} \mathfrak{m}_{P_{a,b}}^{\mu_{a,b}}$ by its definition, we have

$$\dim_{\mathbb{F}} R/J_N = \sum_{P_{a,b}} \dim_{\mathbb{F}} \mathcal{O}_{P_{a,b}} / \mathfrak{m}_{P_{a,b}}^{\mu_{a,b}} = \sum_{1 \leq a \leq q^2, 1 \leq b \leq q} \mu_{a,b}.$$

Hence $\dim_{\mathbb{F}} R/J = \dim_{\mathbb{F}} R/J_N$. Together with $J \subset J_N$, this implies that $J = J_N$. □

Example 7 (continued). We compute generators g_1, g_2 of $J_{N^{(0)}}$. We arrange the points as

$$\begin{aligned} P_{1,1} &= P_2, & P_{1,2} &= P_1, \\ P_{2,1} &= P_4, & P_{2,2} &= P_3, \\ P_{3,1} &= P_6, & P_{3,2} &= P_5, \\ P_{4,1} &= P_7, & P_{4,2} &= P_8, \end{aligned}$$

so that $\mu_{a,b}$ are in decreasing order,

$$\begin{aligned} \mu_{1,1} &= 4, \mu_{1,2} = 3, \\ \mu_{2,1} &= 5, \mu_{2,2} = 3, \\ \mu_{3,1} &= 4, \mu_{3,2} = 2, \\ \mu_{4,1} &= 5, \mu_{4,2} = 2. \end{aligned}$$

We will see in the next subsection that

$$f_{1,2} = \alpha^2 x^7 + \alpha x^6 + \alpha x^4 + x^3 + \alpha^2 x^2 + y + \alpha^2 x + 1$$

satisfies

$$\begin{aligned} v_{P_{1,1}}(y - f_{1,2}) &\geq 1, \\ v_{P_{2,1}}(y - f_{1,2}) &\geq 2, \\ v_{P_{3,1}}(y - f_{1,2}) &\geq 2, \\ v_{P_{4,1}}(y - f_{1,2}) &\geq 3. \end{aligned}$$

Therefore

$$\begin{aligned} g_1 &= (x - 0)^4 (x - 1)^5 (x - \alpha)^4 (x - \alpha^2)^5 = x^{18} + \alpha x^{17} + \alpha^2 x^{16} + x^6 + \alpha x^5 + \alpha^2 x^4, \\ g_2 &= (x - 0)^3 (x - 1)^3 (x - \alpha)^2 (x - \alpha^2)^2 (y - f_{1,2}) \end{aligned}$$

$$= (x^{10} + x^9 + x^4 + x^3)y + \alpha^2 x^{17} + x^{16} + \alpha x^{15} + \alpha x^{14} + \alpha^2 x^{13} + \alpha x^{12} + \alpha^2 x^{11} \\ + \alpha^2 x^{10} + \alpha^2 x^9 + \alpha x^8 + \alpha^2 x^7 + \alpha x^6 + \alpha x^4 + x^3$$

generates $J_{N(0)}$ as a module over $\mathbb{F}[x]$.

D. Computing $y - f_{b,c}$

As $y = x^{q+1} - y^q$, we see that $y = \sum_{i=0}^{\infty} (-1)^i x^{(q+1)q^i}$ in the completion of the local ring at $(0, 0)$. On the other hand, if (α, β) is a rational point of H , then $x \mapsto x - \alpha$, $y \mapsto y - \alpha^q(x - \alpha) - \beta$ defines an automorphism of H taking (α, β) to $(0, 0)$. Hence at (α, β) , we have

$$y = \beta + \alpha^q(x - \alpha) + \sum_{i=0}^{\infty} (-1)^i (x - \alpha)^{(q+1)q^i}. \quad (5)$$

Now we consider the following problem. Suppose that $Q_i = (\alpha_i, \beta_i)$, $1 \leq i \leq r$ are rational points on H with distinct α_i . Given some positive integers μ_i for $1 \leq i \leq r$. We want to construct $y - f$ with $f \in \mathbb{F}[x]$ such that $v_{Q_i}(y - f) \geq \mu_i$ for $1 \leq i \leq r$. There are at least two ways to do this.

First method: For $1 \leq i \leq r$, let w_i be the truncation of the series expansion (5) of y at (α_i, β_i) modulo $(x - \alpha)^{\mu_i}$, and let $s_i, t_i \in \mathbb{F}[x]$ be defined by

$$s_i = \prod_{\substack{j=1 \\ j \neq i}}^r (x - \alpha_j)^{\mu_j} \quad \text{and} \quad s_i t_i \equiv 1 \pmod{(x - \alpha_i)^{\mu_i}}.$$

Then $y - \sum_{i=1}^r w_i s_i t_i$ satisfies the required conditions by the Chinese remainder theorem.

Second method: A somewhat more explicit way is as follows. If $f(x) = \sum_{i=0}^{N-1} a_i x^i \in \mathbb{F}[x]$, then the condition $v_P(y - f) \geq \mu$ is equivalent to the following linear conditions on the coefficients a_i ,

$$\sum_{i=0}^{N-1} \binom{i}{j} \alpha^{i-j} a_i = c_j$$

for $j = 0, 1, \dots, \mu - 1$, where $c_j = 0$ except $c_0 = \beta$, $c_1 = \alpha^q$, $c_{(q+1)q^i} = (-1)^i$ for $i \geq 0$. Now let $N = \mu_1 + \mu_2 + \dots + \mu_r$. Then the required f can be determined by solving the linear system $vA = C$ for the vector $v = (a_0, a_1, a_2, \dots, a_{N-1})$ where C is a certain vector of length N and A is a square matrix of size N obtained by the horizontal join of $N \times \mu_k$ matrices

$$A_k = \left[\binom{i}{j} \alpha_k^{i-j} \right]_{0 \leq i \leq N-1, 0 \leq j \leq \mu_k-1}$$

for $1 \leq k \leq r$. The matrix A is called a confluent Vandermonde matrix in the literature, and is known to be invertible (actually the determinant is $\prod_{i,j} (\alpha_i - \alpha_j)^{\mu_i \mu_j}$ [22], [23]). Therefore the linear system has a unique solution.

Example 8 (continued). Let us compute $f_{1,2}$ in the previous example by the second method. Here $N = 1+2+2+3 = 8$. If $f_{1,2}(x) = \sum_{i=0}^7 a_i x^i$, then $(a_0, a_1, \dots, a_7)A = C$ where

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & \alpha & 1 & \alpha^2 & 1 & 0 \\ 0 & 1 & 0 & \alpha^2 & 0 & \alpha & 0 & 1 \\ 0 & 1 & 1 & 1 & \alpha^2 & 1 & \alpha & \alpha^2 \\ 0 & 1 & 0 & \alpha & 0 & \alpha^2 & 0 & 0 \\ 0 & 1 & 1 & \alpha^2 & \alpha & \alpha & \alpha^2 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & \alpha^2 \\ 0 & 1 & 1 & \alpha & 1 & \alpha^2 & 1 & \alpha \end{bmatrix}$$

and $C = (1, \alpha^2, 1, \alpha^2, \alpha^2, \alpha, \alpha, 0)$. The solution of this linear system was given in the previous subsection.

E. Converting to a Gröbner basis to pick up the Q -polynomial

For this task, we use the Gröbner conversion algorithm in [17] that converts a set of generators of a submodule of $\mathbb{F}[x]^N$ to a module Gröbner basis with respect to a special weighted monomial order. We review the algorithm below.

Let $T = \{(i, j) \mid 0 \leq i \leq l, 0 \leq j \leq q-1\}$. Tuples in T are ordered lexicographically such that $(0, 0)$ is the first tuple in T and the successor of (i, j) is $(i, j+1)$ if $j < q-1$ or $(i+1, 0)$ if $j = q-1$. Thus $\{y^j z^i \mid (i, j) \in T\}$ is a basis for $R[z]_l$ as an $\mathbb{F}[x]$ -module and the weight of the basis element $y^j z^i$ is $ui + (q+1)j$. The index of $f \in R[z]_l$ is defined to be the largest tuple (i, j) such that the coefficient of $y^j z^i$ is nonzero. In particular, if the leading term of $f \in R[z]_l$ is $x^i y^j z^k$ with respect to $>_u$, then $\text{ind}(\text{lt}(f)) = (k, j)$. Note that $\text{ind}(g_{i,j}) = (i, j)$ for the generators $g_{i,j}$ of $I_{M,l}$ computed by Algorithm B.

Algorithm I. The algorithm finds the element of $I_{M,l}$ with the smallest leading term. Initially set $g_{i,j}$ to be the initial set of generators of the module $I_{M,l}$ computed by Algorithm B. Let

$$g_{i,j} = \sum_{(i',j') \in T} a_{i,j,i',j'} y^{j'} z^{i'}$$

during the execution of the algorithm. For $r = (r_1, r_2)$ and $s = (s_1, s_2)$ in T , we abbreviate $a_{r,s} = a_{r_1,r_2,s_1,s_2}$.

11. Set $r \leftarrow (0, 0)$.
12. Set r to the successor of r . If $r \in T$, then proceed; otherwise go to step 16.
13. Set $s \leftarrow \text{ind}(\text{lt}(g_r))$. If $s = r$, then go to step 12.
14. Set $d \leftarrow \deg(a_{r,s}) - \deg(a_{s,s})$ and $c \leftarrow \text{lc}(a_{r,s})\text{lc}(a_{s,s})^{-1}$.
15. If $d \geq 0$, then set

$$g_r \leftarrow g_r - cx^d g_s.$$

If $d < 0$, then set, storing g_s in a temporary variable,

$$g_s \leftarrow g_r, \quad g_r \leftarrow x^{-d} g_r - cg_s.$$

Go back to step 13.

16. Output $g_{i,j}$ with the smallest leading term, and the algorithm terminates.

Example 9 (continued). Algorithm I converts the initial basis given in (3) to a Gröbner basis with respect to the order $>_u$. The computed Gröbner basis is

$$\begin{bmatrix} \alpha x^{10} + \dots & \alpha^2 x^8 + \dots & \alpha x^7 + \dots & x^5 + \dots & \dots & 0 & 0 & 0 & 0 \\ x^{10} + \dots & \alpha x^9 + \dots & \alpha x^8 + \dots & \alpha x^6 + \dots & \dots & 0 & 0 & 0 & 0 \\ \alpha x^{10} + \dots & \alpha x^8 + \dots & \alpha^2 x^8 + \dots & x^6 + \dots & \dots & 0 & 0 & 0 & 0 \\ x^8 + \dots & \alpha^2 x^8 + \dots & x^7 + \dots & x^6 + \dots & \dots & 0 & 0 & 0 & 0 \\ x^9 + \dots & x^7 + \dots & x^6 + \dots & x^5 + \dots & \dots & 0 & 0 & 0 & 0 \\ x^9 + \dots & \alpha^2 x^8 + \dots & \alpha x^7 + \dots & \alpha x^6 + \dots & \dots & 0 & 0 & 0 & 0 \\ \alpha x^8 + \dots & \alpha x^6 + \alpha x^3 & \alpha^2 x^7 + \dots & \alpha^2 x^5 + \dots & \dots & 0 & 0 & 0 & 0 \\ x^8 + \dots & x^6 + \dots & \alpha^2 x^5 + \dots & \alpha^2 x^5 + \dots & \dots & 1 & 0 & 0 & 0 \\ \alpha^2 x^9 + \dots & \alpha^2 x^7 + \dots & x^7 + \dots & \alpha x^5 + \dots & \dots & x & 0 & 0 & 0 \\ x^9 + \dots & \alpha x^7 + \dots & \alpha x^7 + \dots & x^6 + \dots & \dots & 0 & 1 & 0 & 0 \\ \alpha x^{10} + \dots & \alpha x^8 + \dots & x^8 + \dots & \alpha x^6 + \dots & \dots & \alpha^2 x^2 + \alpha^2 x & \alpha^2 & 1 & 0 \\ \alpha^2 x^{11} + \dots & x^{10} + \dots & \alpha x^9 + \dots & x^8 + \dots & \dots & \alpha^2 x^3 & x^2 + \dots & 0 & 1 \end{bmatrix}.$$

The twelve rows represent the polynomials in the Gröbner basis of the module $I_{M,l}$ over $\mathbb{F}[x]$. Comparing the weights of the leading coefficients of the polynomials, which lie on the diagonal, we find that the polynomial represented by the eleventh row is the required Q -polynomial of the ideal I_M , given explicitly in Example 3 equation (1).

V. COMPLEXITY ANALYSIS

Elements of R can be written uniquely as polynomials in y of degree less than q with coefficients in $\mathbb{F}[x]$. We assume that for computations in R , we use this representation of elements of R . Also we think of $\deg_x(f)$ and $\deg_y(f)$ for $f \in R$ in this representation. Note that a straightforward way of multiplying two elements f, g of R takes $O(q^2 ab)$ multiplications on \mathbb{F} and that $\deg_x(fg) \leq a + b + q + 1$ if $a = \deg_x(f)$ and $b = \deg_x(g)$.

First we consider computing $f \in \mathbb{F}[x]$ satisfying $v_{P_i}(y - f) \geq \mu_i$ for $1 \leq i \leq r$ as in Section IV-D. This computation takes $O(N^3)$ multiplications on \mathbb{F} where $N = \mu_1 + \mu_2 + \dots + \mu_r$, if we use Gaussian elimination to solve the linear system. Note also $\deg_x(f) \leq N - 1$.

Next we consider computing g_c according to Proposition 7 in Section IV-C. The first product π_1 on the right side of (4) has at most lq^2 linear factors. Hence π_1 can be computed with $O(l^2 q^4)$ multiplications on \mathbb{F} . Note $\deg_x(\pi_1) \leq lq^2$. On the other hand, as $\deg_x(f_{b,c}) < lq^2$, the second product π_2 can be computed with $O(c^2 l^2 q^4)$ multiplications on \mathbb{F} . Note $\deg_y(\pi_2) \leq c - 1$ and $\deg_x(\pi_2) \leq (c - 1)lq^2$. Then π_1 and π_2 can be multiplied with $O(c^2 l^2 q^4)$ multiplications on \mathbb{F} . Hence, in total, computing g_c takes $O(c^2 l^2 q^4)$ multiplications on \mathbb{F} . Note $\deg_x(g_c) \leq clq^2$ and $\deg_y(g_c) \leq c - 1$.

Now we consider computations in steps B2 and B3 of Algorithm B in Section IV-A. Fix s . Computing η_t ($= g_{t+1}$), as shown above, takes $O((t + 1)^2 l^2 q^4)$ multiplications on \mathbb{F} for each $t = 0, 1, \dots, q - 1$. Computing $h^{(s)}$ can be done with $O(nq^2)$ multiplications on \mathbb{F} . Note $\deg_x(h^{(s)}) \leq q^2 - 1$. Let $\pi^{(s)}$ denote the product of the right side in (2). It is easy to verify $\deg_z(\pi^{(s)}) = s$ and $\deg_x(\pi^{(s)}) \leq s(q^2 - 1) + (s - 1)(q + 1)$ if $s \geq 1$. So computing $g_{s,t} = \eta_t \pi^{(s)}$ takes $O(s^2 t l q^6)$ multiplications on \mathbb{F} . Note $\deg_x(g_{s,t}) \leq t l q^2 + s(q^2 + q)$. Computing $\pi^{(s+1)} = \pi^{(s)}(z - h^{(s)})$ takes $O(s^2 q^6)$ multiplications on \mathbb{F} .

Summing up, an execution of Algorithm B takes

$$\begin{aligned} & \sum_{s=0}^l \left(O(nq^2) + O(s^2 q^6) + \sum_{t=0}^{q-1} O((t+1)^2 l^2 q^4) + \sum_{t=0}^{q-1} O(s^2 t l q^6) \right) \\ &= \sum_{s=0}^l (O(s^2 q^6) + O(l^2 q^7) + O(s^2 l q^8)) \\ &= O(l^3 q^6) + O(l^3 q^7) + O(l^4 q^8) \\ &= O(l^4 q^8) \end{aligned}$$

multiplications on \mathbb{F} . Lastly noting $\deg_u(g_{s,t}) = O(lq^4)$ and using a result in [17], we see that an execution of Algorithm I takes $O(l^5 q^{10})$ multiplications on \mathbb{F} . Therefore the algebraic soft-decision decoder of Hermitian codes can be implemented in a way that takes $O(l^5 q^{10}) = O(l^5 n^{3+1/3})$ multiplications on \mathbb{F} .

VI. SIMULATION RESULTS

We implemented the algebraic soft-decision decoder (SDD) for Hermitian codes in software. In this section, we present some simulation results that show the performance of the algebraic soft-decision decoder for half-rate Hermitian codes.

First we describe the general setup of our simulations. We assume the AWGN channel. For QPSK and QAM modulations, the signal points correspond one-to-one with the symbols in the finite field over which the code is defined, and the posterior probabilities of the symbols are simply set to those of the corresponding signal points. For BPSK, each of the symbols correspond with a bit sequence, and the posterior probabilities of the symbols are set to the products of the posterior probabilities of the bits. Koetter and Vardy's multiplicity assignment algorithm [3] is used to translate the posterior probabilities to the values of the multiplicity matrix. The multiplicity assignment algorithm accepts a parameter L that limits the z -degree of the Q -polynomial, thereby the list size of the candidate codewords to at most L . From the multiplicity matrix, our interpolation algorithm finds the Q -polynomial. Then Wu and Siegel's root-finding algorithm [16] is used to compute the roots of the Q -polynomial. The list of candidate codewords is then formed from the roots. If the list is empty, then the decoder simply output the message part of the received vector determined by hard-decision directly from the posterior probabilities of the symbols. If the list is not empty, the decoder outputs the message from the codeword that has the best score with respect to the multiplicity matrix.

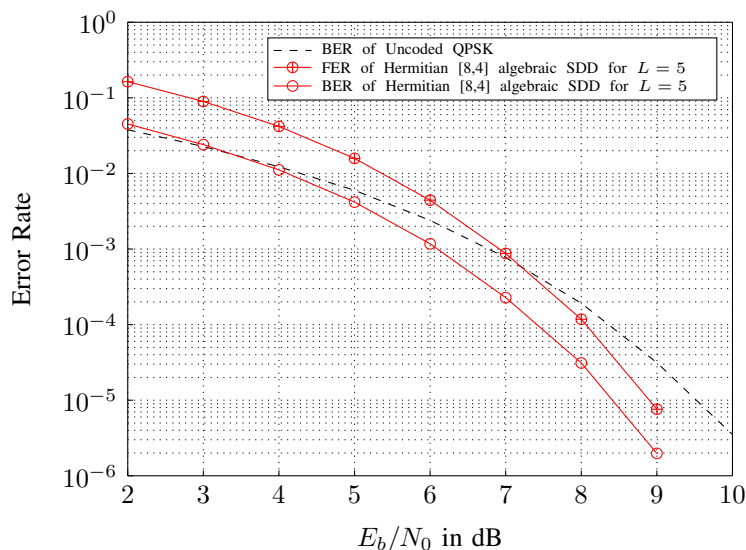


Fig. 1. Performance of algebraic SDD of $[8, 4]$ Hermitian code over \mathbb{F}_4 with QPSK modulation.

A. $[8, 4]$ Hermitian code with QPSK

The smallest field over which Hermitian codes are defined is \mathbb{F}_4 and the length of these codes is 8. The length is extremely small, and it is perhaps unrealistic to expect the codes to be used in practice. However the codes are amenable for simulations with somewhat larger L . Figure 1 show the performance of the half-rate $[8, 4]$ Hermitian code with QPSK. The example used in previous sections was sampled from this simulation with SNR 2 and $L = 5$.

B. $[64, 32]$ Hermitian code with BPSK

Figures 2 and 3 show the performance of $[64, 32]$ Hermitian code with BPSK modulation. For comparison, the figures also show the performance of the half-rate $[16, 8]$ Reed-Solomon code. Observe that the performance curve of Hermitian code more steeply decrease than that of Reed-Solomon code, and from around 5 dB, the Hermitian code outperforms the Reed-Solomon code.

C. $[64, 32]$ Hermitian code with 16-QAM

Figures 4 and 5 also show that the Hermitian code outperforms the Reed-Solomon code with 16-QAM modulation, from around 8 dB onward.

VII. CONCLUSION

We presented an algebraic soft-decision decoder for Hermitian codes. Software simulations show that Hermitian codes perform better than Reed-Solomon codes for algebraic soft-decision decoding, as expected. However, for the decoder to be really practical, reduction of the computational complexity remains an important problem. One promising avenue is to generalize the idea of complexity reduction for Reed-Solomon codes in [9]. Designing efficient electric circuits implementing the decoder is of course an issue to explore.

The extent of our computer simulations of the decoding algorithm was limited by our computing resources. It would be good to have analytic results about the performance of the decoding algorithm. There have been several analytic performance analyses for the algebraic soft-decision decoding of Reed-Solomon codes [8]. Similar analyses may be done for Hermitian codes.

Our description of the decoding algorithm is interwoven with the particular structure of Hermitian codes. However, the underlying principle of the decoding algorithm seems to apply to a wider class of algebraic geometry codes. In particular, plane algebraic curves with one point at infinity are immediate candidates. We leave an adequate treatment of this subject as a remaining work.

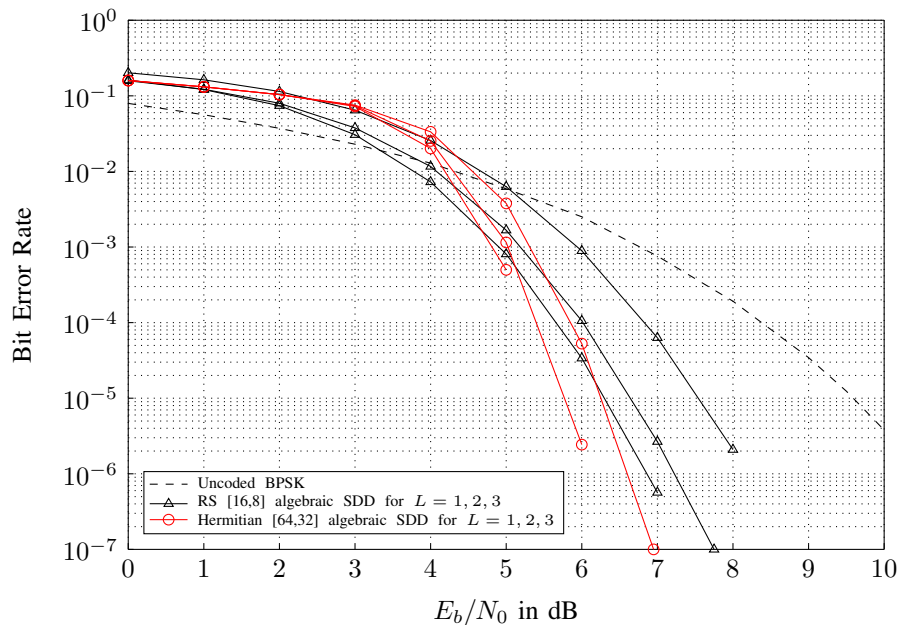


Fig. 2. Bit Error Performance of algebraic SDD of $[64, 32]$ Hermitian code over \mathbb{F}_{16} with BPSK modulation.

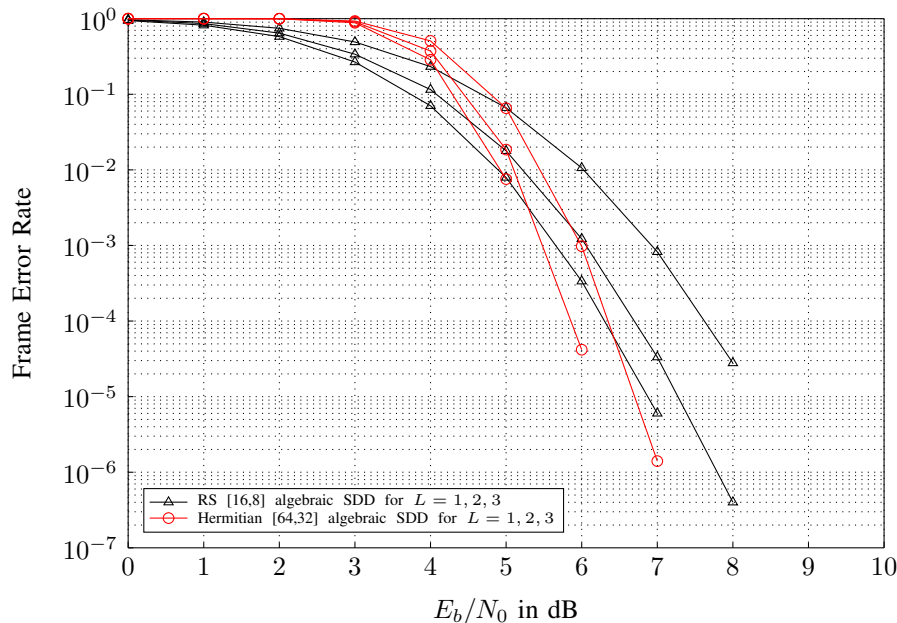


Fig. 3. Frame Error Performance of algebraic SDD of $[64, 32]$ Hermitian code over \mathbb{F}_{16} with BPSK modulation.

REFERENCES

- [1] M. Sudan, "Decoding of Reed-Solomon codes beyond the error-correction bound," *J. Complexity*, vol. 13, no. 1, pp. 180–193, 1997.
- [2] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, 1999.
- [3] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, 2003.
- [4] —, "A complexity reducing transformation in algebraic list decoding of Reed-Solomon codes," in *Proc. IEEE Symp. Information Theory Workshop*, Paris, France, Apr. 2003, pp. 10–13.
- [5] V. Olshevsky and M. A. Shokrollahi, "A displacement approach to decoding algebraic codes," in *Fast algorithms for structured matrices: theory and applications*, ser. Contemp. Math. Amer. Math. Soc., 2003, vol. 323, pp. 265–292.

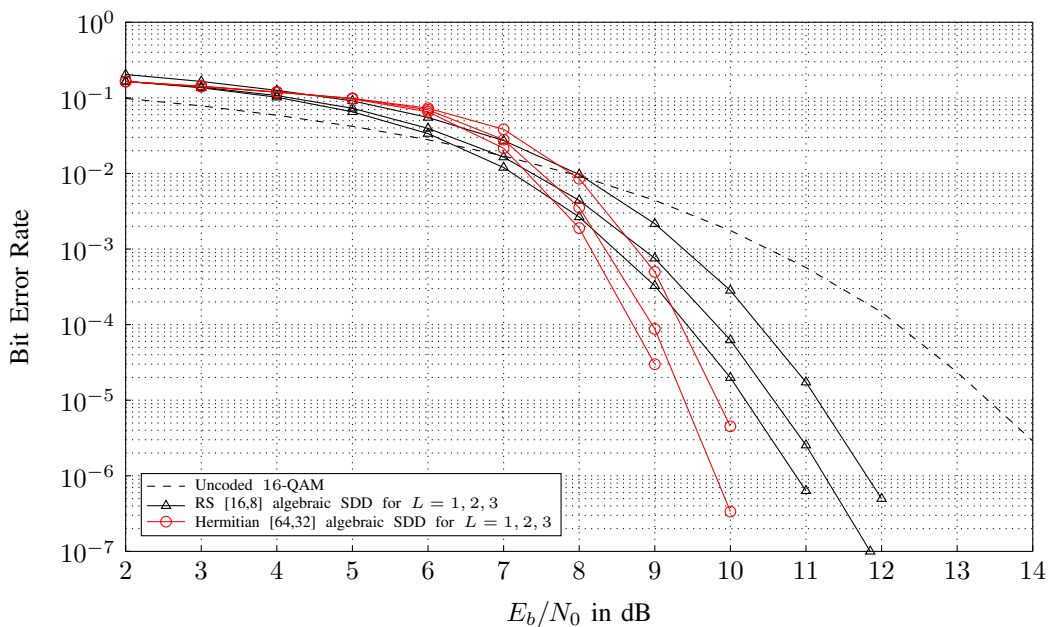


Fig. 4. Bit Error Performance of algebraic SDD of $[64, 32]$ Hermitian code with 16-QAM modulation.

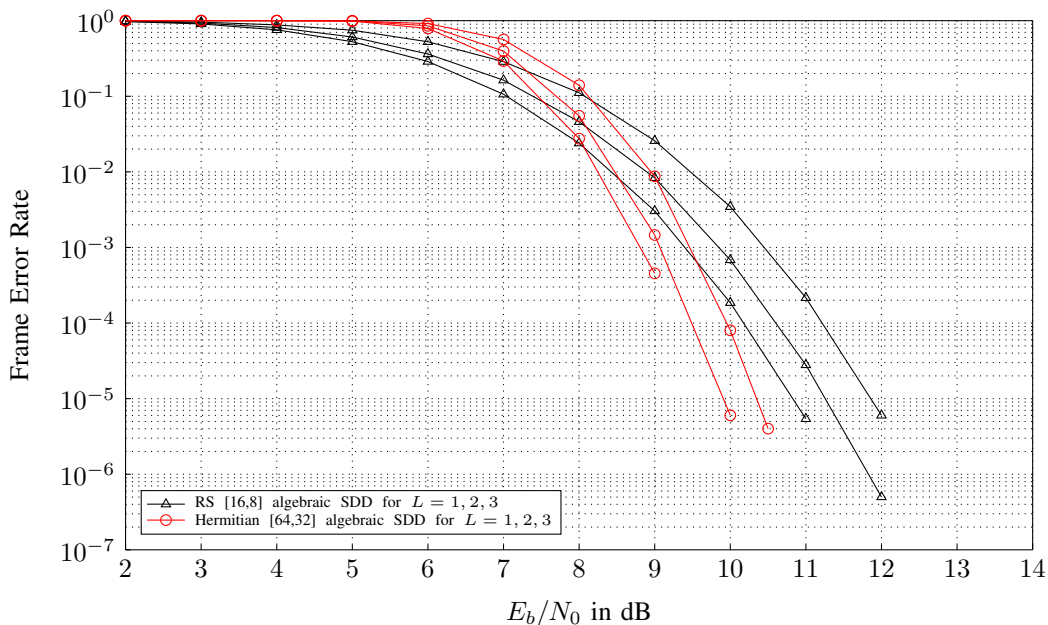


Fig. 5. Frame Error Performance of algebraic SDD of $[64, 32]$ Hermitian code with 16-QAM modulation.

- [6] M. Kuijper and J. W. Polderman, "Reed-Solomon list decoding from a system-theoretic perspective," *IEEE Trans. Inf. Theory*, vol. 50, no. 2, pp. 259–271, 2004.
- [7] M. Alekhovich, "Linear Diophantine equations over polynomials and soft decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2257–2265, 2005.
- [8] N. Ratnakar and R. Koetter, "Exponential error bounds for algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3899–3917, 2005.
- [9] J. Ma and A. Vardy, "A complexity reducing transformation for the Lee-O'Sullivan interpolation algorithm," in *Proc. IEEE Symp. Information Theory*, Nice, France, Jun. 2007.
- [10] A. Ahmed, R. Koetter, and N. R. Shanbha, "VLSI architectures for soft-decision decoding of Reed-Solomon codes," in *Proc. IEEE Conf. Communications*, vol. 5, Jun. 2004, pp. 2584–2590.
- [11] J. Ma, A. Vardy, and Z. Wang, "Efficient fast interpolation architecture for soft-decision decoding of Reed-Solomon codes," in *Proc. IEEE*

- Symp. Circuits and Systems*, Kos, Greece, May 2006, pp. 4823–4826.
- [12] J. Ma, A. Vardy, Z. Wang, and Q. Chen, “Direct root computation architecture for algebraic soft-decision decoding of Reed-Solomon codes,” in *Proc. IEEE Symp. Circuits and Systems*, New Orleans, LA, May 2007, pp. 1409–1412.
 - [13] W. J. Gross, F. R. Kschischang, R. Koetter, and G. Gulak, “Applications of algebraic soft-decision decoding of Reed-Solomon codes,” *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 1224–1234, 2006.
 - [14] H. Stichtenoth, *Algebraic Function Fields and Codes*. Springer-Verlag, 1993.
 - [15] M. A. Shokrollahi and H. Wasserman, “List decoding of algebraic-geometric codes,” *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 432–437, 1999.
 - [16] X.-W. Wu and P. H. Siegel, “Efficient root-finding algorithm with applications to list decoding of algebraic-geometric codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 6, pp. 2579–2587, 2001.
 - [17] K. Lee and M. E. O’Sullivan, “List decoding of Hermitian codes using Gröbner bases,” *arXiv:cs/0610132*, Oct. 2006.
 - [18] —, “An interpolation algorithm using Gröbner bases for soft-decision decoding of Reed-Solomon codes,” in *Proc. IEEE Symp. Information Theory*, Seattle, WA., Jul. 2006, pp. 2032–2036.
 - [19] W. Fulton, *Algebraic Curves*. Benjamin, 1969.
 - [20] D. Cox, J. Little, and D. O’Shea, *Using Algebraic Geometry*, ser. GTM. Springer-Verlag, New York, 1998, vol. 185.
 - [21] M. F. Atiyah and I. G. MacDonald, *Introduction to commutative algebra*. Perseus Books, 1969.
 - [22] C. Krattenthaler, “Advanced determinant calculus,” *Sém. Lothar. Combin.*, vol. 42, pp. Art. B42q, 67 pp. (electronic), 1999, the Andrews Festschrift (Maratea, 1998).
 - [23] S.-H. Hou and W.-K. Pang, “Inversion of confluent vandermonde matrices,” *Computers and Mathematics with Applications*, vol. 43, pp. 1539–1547, 2002.